PATENT
Attorney Docket No. 401378

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

YOSHIYUKI HARAGUCHI

Application No.	Unassigned	Art Unit:	Unassigned
Filed:	September 21, 2001	Examiner:	Unassigned
For:	MICROPROCESSOR		

PRELIMINARY AMENDMENT

Commissioner for Patents
Washington, D. C. 20231

Dear Sir:

Prior to the examination of the above-identified patent application, please enter the following amendments and consider the following remarks.

IN THE SPECIFICATION:

Replace the paragraph beginning at page 1, line 12, with:

It is known that the method of pre-fetching instructions increases processing speed of a microprocessor. A group of instructions that have been stored in sequential addresses are sequentially executed in ordinary sequential computers. In contrast, in the instruction pre-fetch system, an instruction located several instructions ahead, which is expected to be used in the future, is taken out in parallel with the executing and decoding processes of the previous instruction.

Replace the paragraph beginning at page 1, line 21, with:

In other words, an instruction, which has been preliminarily pre-fetched from a main memory or a cache, is stored in an instruction pre-fetch buffer (queue buffer) with a small capacity that enables high-speed access; thus, an attempt is made to virtually reduce a delay in the execution caused by memory access at the time of the instruction fetch.

Replace the paragraph beginning at page 2, line 18, with:

In Japanese Patent Application Laid-Open No. 7-73034, a comparison is made between the branch end address at the time of executing a branch instruction and the corresponding address range of the instruction located in the queue buffer, and when the branch end address is located within the corresponding address range, the instruction in the queue buffer is used without flushing the queue buffer, thereby making it possible to reduce the number of pre-fetches after the branch.

Replace the paragraph beginning at page 3, line 2, with:

In this conventional technique, the number of pre-fetches after the branch is certainly reduced; however, since the branch instruction is dealt with as a normal non-conditional branch instruction, a complex address generation process is required for a branch after decoding the instruction, and the corresponding circuits become complex and bulky.

Replace the paragraph beginning at page 6, line 1, with:

Fig. 1 is a block diagram that shows an example of an internal construction of a microprocessor in accordance with this invention;

IN THE CLAIMS:

Replace the indicated claims with:

1. (Amended) A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and

a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from the program counter, wherein, when the instruction decoder recognizes reception of a branch instruction, the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as an operand of the branch instruction, outputs an instruction word length of the branch instruction including the operand to the program counter, thereby updating the address of the program counter, and prevents flushing of the queue buffer.

2. (Amended) The microprocessor according to claim 1, wherein a label is used to specify the branch end.

3. (Amended) The microprocessor according to claim 1, wherein a relative address between the branch instruction and branch end is specified to specify the branch end.

4. (Amended) The microprocessor according to claim 3, wherein

the queue controller controls input and output of instructions to the queue buffer so that a plurality of previous instructions, which correspond to a number of relative addresses from the instruction that is currently being executed, remain in the queue buffer, and

a minus relative address may be specified as the relative address.

5. (Amended) A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from the program counter, wherein, when the instruction decoder recognizes reception of a branch instruction, the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as NOP instructions, outputs an instruction word length corresponding to the branch instruction and the NOP instructions to the program counter, thereby updating the address of the program counter, and prevents flushing of the queue buffer.

IN THE ABSTRACT:

Replace the abstract with:

ABSTRACT OF THE DISCLOSURE

A microprocessor avoids loss of instructions in a pre-fetch procedure when a branch instruction is received. When a new branch instruction that specifies a branch end is received by a queue buffer, all the instructions preceding the specified branch end are processed as an operand of the branch instruction. Moreover, the instruction word length of the branch instruction including the instruction that has been processed as the operand is output to a program counter, so the queue buffer is not flushed.

REMARKS

The foregoing amendments are made to correct minor translational errors and to meet United States requirements as to form. No new matter is added.

Respectfully submitted,

LEYDIG, VOIT & MAYER, LTD.

Jeffrey A. Wyand,
Registration No. 29,458

Suite 300
700 Thirteenth Street, N. W.
Washington, D. C. 20005
Telephone: (202) 737-6770
Facsimile: (202) 737-6776
Date: Sept 21, 2001
JAW:cmcg

PATENT
Attorney Docket No. 401378

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

YOSHIYUKI HARAGUCHI

Application No.	Unassigned	Art Unit:	Unassigned
Filed:	September 21, 2001	Examiner:	Unassigned
For:	MICROPROCESSOR		

AMENDMENTS TO SPECIFICATION, CLAIMS, AND ABSTRACT MADE VIA PRELIMINARY AMENDMENT

Amendments to the paragraph beginning at page 1, line 12:

It is known that the method of pre-fetching ~~the~~ instructions increases processing speed of a microprocessor. A group of instructions that have been stored in sequential addresses are sequentially executed in ~~the~~ ordinary sequential computers. In contrast, in the instruction pre-fetch system, an instruction located several instructions ahead, which is expected to be used in the future, is taken out in parallel with the executing and decoding processes of the previous instruction.

Amendments to the paragraph beginning at page 1, line 21:

In other words, an instruction, which has been preliminarily pre-fetched from a main memory or a ~~cash~~ cache, is stored in an instruction pre-fetch buffer (queue buffer) with a small capacity that enables ~~a~~ high-speed access; thus, an attempt is made to virtually reduce a delay in the execution caused by ~~a~~ memory access at the time of the instruction fetch.

Amendments to the paragraph beginning at page 2, line 18:

~~There, in~~In Japanese Patent Application Laid-Open No. 7-73034, a comparison is made between the branch end address at the time of executing a branch instruction and the corresponding address range of the instruction located in the queue buffer, and when the branch end address is located within the corresponding address range, the instruction in the queue buffer is used without flushing the queue buffer, thereby making it possible to reduce the number of pre-fetches after the branch.

Amendments to the paragraph beginning at page 3, line 2:

In this conventional technique, ~~although~~ the number of pre-fetches after the branch is certainly reduced; however, since the ~~brand~~ branch instruction is dealt with as a normal non-conditional branch instruction, a complex address generation process is required for a branch after decoding the instruction, and the corresponding circuits ~~becomes~~ become complex and bulky.

Amendments to the paragraph beginning at page 6, line 11:

Fig. 1 is a block diagram that shows an example of an ~~inner~~ internal construction of a microprocessor in accordance with this invention;

Amendments t the existing claims:

1. (Amended) A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address ~~on~~ in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and

a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from ~~said~~ the program counter, wherein, when ~~said~~ the instruction decoder recognizes reception of a ~~predetermined~~ branch instruction, ~~it~~ the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as an operand of the branch instruction, outputs an instruction word length of the branch instruction including the operand to the program counter, thereby updating the address of the program counter, and ~~provides a control so as not to flush~~ prevents flushing of the queue buffer.

2. (Amended) The microprocessor according to claim 1, wherein a label is used ~~so as~~ to specify the branch end.

3. (Amended) The microprocessor according to claim 1, wherein a relative address between the branch instruction and branch end is specified ~~so as~~ to specify the branch end.

4. (Amended) The microprocessor according to claim 3, wherein

~~said~~ the queue controller ~~carries out the~~ controls input and output ~~control~~ of instructions to the queue buffer so that a plurality of ~~the~~ previous instructions, which correspond to a ~~predetermined~~ number of relative addresses from the instruction that is currently being executed, ~~are allowed to~~ remain in the queue buffer, and

a minus relative address may be specified as the relative address.

5. (Amended) A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address ~~on~~ in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and

a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from ~~said~~ the program counter, wherein, when ~~said~~ the instruction decoder recognizes reception of a ~~predetermined~~ branch instruction, ~~it~~ the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as NOP instructions, outputs an instruction word length corresponding to the branch instruction and the NOP instructions to the program counter, thereby updating the address of the program counter, and ~~provides a control so as~~

~~not to flush~~ prevents flushing of the queue buffer.

Amendments to the abstract:

ABSTRACT OF THE DISCLOSURE

A microprocessor avoids loss of instructions in a pre-fetch procedure when a branch instruction is received. When a new branch instruction that specifies a branch end ~~is added. When this branch instruction~~ is received by a queue buffer, all the instructions preceding the specified branch end are processed as an operand of the branch instruction. Moreover, the instruction word length of the branch instruction including the instruction that has been processed as the operand is output to ~~the~~ a program counter, ~~and~~ so the queue buffer is not flushed.

PATENT
Attorney Docket No. 401378

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

YOSHIYUKI HARAGUCHI

Application No.	Unassigned	Art Unit:	Unassigned
Filed:	September 21, 2001	Examiner:	Unassigned
For:	MICROPROCESSOR		

PENDING CLAIMS AFTER ENTRY OF PRELIMINARY AMENDMENT

1. A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and

a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from the program counter, wherein, when the instruction decoder recognizes reception of a branch instruction, the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as an operand of the branch instruction, outputs an instruction word length of the branch instruction including the operand to the program counter, thereby updating the address of the program counter, and prevents flushing of the queue buffer.

2. The microprocessor according to claim 1, wherein a label is used to specify the branch end.

3. The microprocessor according to claim 1, wherein a relative address between the branch instruction and branch end is specified to specify the branch end.

4. The microprocessor according to claim 3, wherein

the queue controller controls input and output of instructions to the queue buffer so that a plurality of previous instructions, which correspond to a number of relative addresses from the instruction that is currently being executed, remain in the queue buffer, and

a minus relative address may be specified as the relative address.

5. A microprocessor comprising:

a main memory which stores instructions;

a queue buffer which pre-fetches and stores instructions from the main memory;

a program counter which generates an address in the main memory in which an instruction to be next executed is stored;

an instruction decoder which receives and decodes instructions output from the queue buffer; and

a queue controller which controls input and output of instructions to the queue buffer based on the address generated and output from the program counter, wherein, when the instruction decoder recognizes reception of a branch instruction, the instruction decoder processes all the instructions preceding a branch end specified by the branch instruction as NOP instructions, outputs an instruction word length corresponding to the branch instruction and the NOP instructions to the program counter, thereby updating the address of the program counter, and prevents flushing of the queue buffer.